



# Image Compression and Deblurring

**Anastase Nakassis**

Advanced Networking Technologies Division

**Alfred Carasso**

Mathematical and Computational  
Sciences Division

U.S. DEPARTMENT OF COMMERCE  
Technology Administration  
Information Technology Laboratory  
National Institute of Standards  
and Technology  
Gaithersburg, MD 20899

QC  
100  
.U56  
NO. 6521  
2000

**NIST**

**National Institute of  
Standards and Technology**  
Technology Administration  
U.S. Department of Commerce



# **Image Compression and Deblurring**

**Anastase Nakassis**

Advanced Networking Technologies Division

**Alfred Carasso**

Mathematical and Computational  
Sciences Division

U.S. DEPARTMENT OF COMMERCE  
Technology Administration  
Information Technology Laboratory  
National Institute of Standards  
and Technology  
Gaithersburg, MD 20899

May 2000



U.S. DEPARTMENT OF COMMERCE  
William M. Daley, Secretary

TECHNOLOGY ADMINISTRATION  
Dr. Cheryl L. Shavers, Under Secretary  
of Commerce for Technology

NATIONAL INSTITUTE OF STANDARDS  
AND TECHNOLOGY  
Raymond G. Kammer, Director



# Image Compression and Deblurring

Anastase Nakassis

Advanced Networking Technologies Division

and

Alfred Carasso

Mathematical and Computational Sciences Division

Information Technology Laboratory

National Institute of Standards and Technology

Gaithersburg, MD 20899

## Abstract

What follows is the description of an experiment in which we investigated the possibility that blurring techniques and lossless compression could be combined as an alternative to lossy compression techniques for still images. Our results show that while blurred images compress losslessly better than their originals, this technique offers relatively modest compression ratios that can be matched by lossily compressed images (jpeg) having less noticeable artifacts.

## 1 Introduction

The dominant image compression techniques are derived from signal representation techniques (e.g., the Fourier transform) and variants thereof (e.g., wavelets). Such techniques operate alternatively on the rows and columns of the image to be processed and in the end represent the image as a sum of sinusoidal functions (or wavelets).

Lossy compression is achieved by dropping the terms that lack visual significance, reducing the precision of the remaining terms as applicable, and losslessly compressing the resultant terms. Effective compression throws away as much precision as possible while retaining acceptable quality levels. Nevertheless, we note that:

- Edges within an image are significant.
- If too many terms are dropped, then the edges are not sharp and blurring occurs; in addition, if the filters used are not carefully chosen, ringing effects are observed (i.e., the reconstructed image contains homothetically displaced echoes of the edges). Both artifacts relate to what is known as the Gibbs phenomenon (see below).

Therefore, acceptable compression presupposes careful choices (filter selection and suppression of information) to avoid the blurring and ringing effects near edges; this precision is redundant over the portions of the image that are smooth. This has led researchers to consider schemes (see [10],[17]) that operate in two steps:

- The first step extracts sufficient information to represent the edges; this information is *subtracted* from the image to be compressed and the result of the *subtraction* is a smooth image;

- the resultant smooth image can be greatly compressed without undesirable side effects.

In the experiment we designed we investigated a different approach. Blurred images lack strong edges and there exist efficient procedures for blurring and deblurring images (see [5],[11],[13]). It is therefore conceivable that one could operate as follows:

- Blur the original image.
- Compress the resultant blurred image.
- Transmit/store the compressed blurred image and have the user deblur it.

We observe nevertheless that:

- blurring introduces artifacts; (more precisely: Artificially blurred images can be fully deblurred if sufficient information is retained; sufficient information with the techniques at hand translates to more bits than we can reasonably afford and, therefore, blurring followed by truncation to the most significant eight bits introduces artifacts).
- deblurring does not tolerate errors well (they tend to be magnified);
- the user must have the software and the computing power necessary to deblur.

In short, the blur-compress technique incurs penalties whose extent must be investigated and, even if it were superior to traditional compression (e.g., jpeg), it would only be practical in constrained environments (low grade communication and/or storage facilities but sufficient computing power).

The remaining sections of the paper provide a summary of the difficulties of efficiently representing discontinuous functions through Fourier series, an overview of blurring and deblurring algorithms, and the data we collected over eight gray scale images.

## 2 Fourier series, discontinuities, and the Gibbs Phenomenon

Infinite sums of the form

$$a_0/2 + \sum_{k=1}^{\infty} (a_k \cos(kx) + b_k \sin(kx)) \quad (1)$$

appear as early as 1753 ([6]) in the solution by D. Bernoulli of the problem of the vibrating string and continue being used by others [Lagrange, D'Alembert, Euler, Clairaut] as tools in the solution of problems that arose on partial differential equations and in astronomy. Nevertheless, they are used in a limited way in situations in which the existence of the series can be demonstrated by ad hoc means. Fourier is the first to state that an arbitrary function could



be expressed as a trigonometric series (1807 and 1811, "Théorie Analytique de la Chaleur", published in 1822).

Fourier's proofs left something to be desired and the first rigorous treatment of the subject is attributed to Dirichlet who proved that if  $f(x)$  is bounded over  $(-\pi, \pi)$  and piecewise monotonic, then its Fourier series converges at each point  $x$  in  $(-\pi, \pi)$  to  $(f(x+0) + f(x-0))/2$  and, in addition, for  $x = \pi$  and for  $x = -\pi$  the series converges to  $(f(-\pi+0) + f(\pi-0))/2$ .

We note that by definition  $f(x+0)$  and  $f(x-0)$  are, respectively, the limits of  $f(x+h)$  and  $f(x-h)$  as  $h$  tends to 0 from above, i.e.,  $h > 0$ . We also note that for ease of exposition and without adverse effect, we may assume that  $f$  is defined over the real numbers, is periodic with period  $2\pi$ , and satisfies  $f(x) = (f(x+0) + f(x-0))/2$  for all  $x$ ; these assumptions imply that if  $0 < h < 2\pi$ , then  $f(\pi+h) = f(-\pi+h)$ ,  $f(\pi-h) = f(-\pi-h)$ , and that  $f(-\pi) = f(\pi) = (f(-\pi+0) + f(\pi-0))/2$ .

Dirichlet's results have since been extended but they are adequate for a great number of applications and what matters from our point of view is the convergence of the series that can be summarized as follows:

Let

$$S_n(x) = a_0/2 + \sum_{k=1}^{k=n} (a_k \cos(kx) + b_k \sin(kx)). \quad (2)$$

Then,

- For every  $x$ ,  $S_n(x)$  converges to  $f(x)$  as  $n \rightarrow \infty$ .
- The convergence is uniform over every closed interval  $[a,b]$  that does not contain points of discontinuity.
- If  $\xi$  is a point of discontinuity, then  $S_n(x)$  oscillates around  $f(x)$  both before and after  $\xi$ . The width of the oscillatory terms tends to 0 as  $n$  increases and they are dampened out as we move away from  $\xi$ . Nevertheless the height of the oscillatory terms that occur immediately before and after  $\xi$  is insensitive to  $n$  and roughly equal to 9% of the jump of  $f(x)$  at  $\xi$ .

The graphical implications of the convergence of  $S_n(x)$  were observed by Albert Michelson who constructed in 1898 a harmonic analyzer capable of determining the first 80 components of the Fourier series and of synthesizing them to produce  $S_n(x)$  ( $n \leq 80$ ) for graphically defined functions  $f$ . Experimentation showed that for most functions  $f(x)$ , the graph of  $S_n(x)$  matched the graph of  $f(x)$  with high degree of precision. Nevertheless, if  $f(x)$  were a square wave (i.e., a step-function) as shown figure 1, then the resultant plot would extend the vertical lines as in figure 2.

Michelson considered the possibility that the extensions of the vertical segments might be due to defects in his equipment and communicated his observations to Josiah Gibbs. The latter explained the phenomenon in a publication in 1899 (the Gibbs phenomenon which had already



Figure 1: A square wave

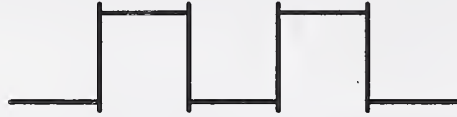


Figure 2: Limit graph of the Fourier series of a square wave

been observed by Wilbraham in 1848 ([18], page 61). As he put it, at the discontinuity points the limit of the graph of  $S_n(x)$  is not equal to the graph of the limit of  $S_n(x)$  (i.e., the graph of  $f(x)$ ). That is, while  $S_n(x)$  converges pointwise to  $f(x)$ , the graph of  $S_n(x)$  does not converge to the graph of  $f(x)$  around points of discontinuity.

In short, near the discontinuity of the step function  $S_n(x)$  looks, roughly, as in figure 3.

The sketch in figure 3 does not purport to be accurate. It does nevertheless illustrate the following:

- Let  $M$  and  $m$  represent, respectively, the maximum and minimum value in  $\{f(\xi+0), f(\xi-0)\}$  and let  $D = M - m$ . If  $f$  is discontinuous at  $\xi$  then  $S_n(x)$  overshoots  $M$  by an amount roughly equal to  $0.09D$  and undershoots  $m$  by the same amount.
- The height of the lobes that  $S_n(x)$  exhibits immediately before and after  $\xi$  is insensitive to the actual value of  $n$  while the widths of the lobes tend to 0 as  $n \rightarrow +\infty$  and, therefore, the lobes extend the vertical wall of the jump as illustrated above).

The practical consequences of the the Gibbs phenomenon are these:

- It impacts filter design because the lobes represent error that typically shows as an echo (audio) or as a ghost image.

To illustrate the visual impact of ringing effects we include figures 4 and 5 which show, respectively, an original image, Barbara, and the same image after some frequencies were suppressed. The degraded image shows significant impairments including blurring and ringing artifacts, most notably at the right elbow of Barbara and along the table's leg.

Note: To obtain the degraded image we used the FBI fingerprint wavelet to decompose the image as follows:



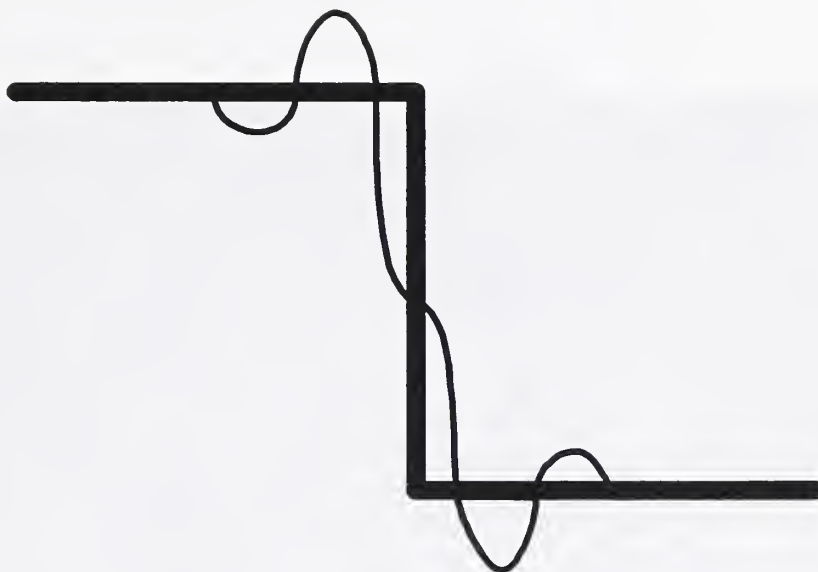


Figure 3:  $S_n(x)$  near a discontinuity point

- By definition,  $Q_6$  is the original image.
  - for all  $k$ ,  $1 \leq k \leq 5$ ,  $Q_k$  and  $R_k$  are respectively the LL band and the LH+HL+HH bands obtained from the wavelet decomposition of  $Q_{k+1}$  via the FBI fingerprint wavelet.
  - By definition  $R_0 = Q_1$ .
  - The impaired image was reconstructed after the coefficients in  $R_3$  and  $R_4$  were zeroed.
- It impacts compression.

Indeed, in the presence of discontinuities the coefficients  $a_n$  and  $b_n$  in the Fourier series converge to 0 no faster than  $1/n$ . The same phenomenon occurs, albeit locally, when wavelets are used. Compression typically works by dropping as much information as possible because it is either numerically too small or because it does not materially affect human perception. Fast convergence to 0 is therefore highly desirable and the rate of convergence relates to the smoothness of the approximated function (i.e., number of continuous derivatives available) [[12],[16]].

For our purposes all this highly summarized discussion boils down to this: if our images were edge free, we could have at the same time high compression rates and high quality.



Figure 4: Original Barbara





Figure 5: Barbara minus some middle frequencies

### 3 Image blurring, Fourier transforms and Lévy kernels

An effective way of blurring an image  $f(x, y)$  so as to smooth out edges and other sharp features, is through convolution with an appropriate smoothing kernel  $h(x, y)$ ,

$$\int_{\mathbb{R}^2} h(x-u, y-v) f(u, v) du dv \equiv h(x, y) \otimes f(x, y) = g(x, y), \quad (3)$$

where  $g(x, y)$  is the resulting blurred image and  $\otimes$  denotes convolution. The prime example of such a smoothing kernel is the Gaussian probability density

$$h_1(x, y) = (\pi a)^{-1} e^{-(x^2+y^2)/a}, \quad a > 0. \quad (4)$$

Another example is the Cauchy or Lorentzian probability density

$$h_2(x, y) = (a/2\pi)(a^2 + x^2 + y^2)^{-3/2}, \quad a > 0. \quad (5)$$

In both cases,  $a > 0$  is a constant that controls the width of the blurring kernel. As  $a$  increases, the width increases and blurring becomes more severe. As  $a \downarrow 0$ , the kernels approach the Dirac  $\delta$ -function and little or no smoothing takes place. To gain a fuller understanding of such smoothing, it is helpful to use the Fourier transform

$$\mathcal{F}[v] = \int_{\mathbb{R}^2} v(x, y) e^{-2\pi i(x\xi + y\eta)} dx dy = \hat{v}(\xi, \eta). \quad (6)$$

One advantage of Fourier methods is that convolution in the space variables corresponds to multiplication in the transform variables. Thus, from (3)

$$\mathcal{F}[h \otimes f] = \hat{h}(\xi, \eta) \hat{f}(\xi, \eta) = \hat{g}(\xi, \eta). \quad (7)$$

Another advantage is that a wide class of smoothing kernels  $h(x, y)$  have a simple functional form in Fourier space. This is the case for the class of symmetric ‘stable’ Lévy probability densities defined by [7],

$$\hat{h}(\xi, \eta) = e^{-\alpha(\xi^2 + \eta^2)^\beta}, \quad \alpha > 0, \quad 0 < \beta \leq 1. \quad (8)$$

Here,  $\alpha$  and  $\beta$  are constants. The constant  $\alpha > 0$  controls the width of the kernel  $h(x, y)$  in real space, and blurring becomes more severe as  $\alpha$  increases. The case  $\beta = 1$  corresponds to the Gaussian kernel (4) while the Cauchy kernel (5) is the case  $\beta = 1/2$ . For other values of  $\beta$ ,  $h(x, y)$  is not known in closed form.

From (7) and (8), it is evident that convolution with such Lévy densities produces attenuation of high frequency components in  $f(x, y)$  through multiplication by a negative exponential in the frequency domain. The constant  $\beta$  also plays an important role. For the same fixed value of  $\alpha > 0$ , high frequency attenuation is more severe for higher values of  $\beta$  than it is for lower values of  $\beta$ . A simple recipe for smoothing an image is to Fourier transform the image, multiply the transformed image by  $\hat{h}(\xi, \eta)$  in (8) with preselected values of  $\alpha$  and  $\beta$ , and then inverse transform to obtain the smoothed version  $g(x, y)$ .

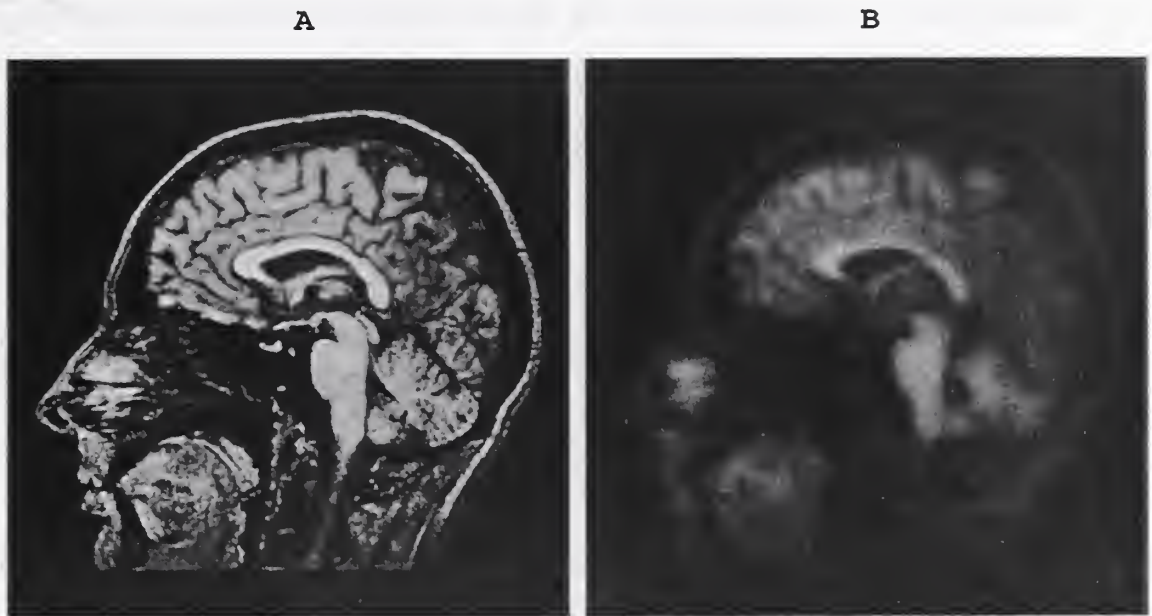


Figure 6: Blurring of MRI brain image by convolution with Cauchy density

- A. Sharp MRI brain image  $f(x, y)$   
 B. Blurred MRI image  $g_s(x, y)$  stored in 8-bit precision

Since  $f(x, y) \geq 0$ , we have

$$|\hat{f}(\xi, \eta)| \leq \int_{R^2} f(x, y) dx dy = \hat{f}(0, 0) = \sigma > 0. \quad (9)$$

Also, since  $h(x, y) \otimes f(x, y) = g(x, y)$  and  $h(x, y)$  is a probability density,

$$|\hat{g}(\xi, \eta)| \leq \hat{g}(0, 0) = \int_{R^2} g(x, y) dx dy = \int_{R^2} f(x, y) dx dy = \sigma > 0. \quad (10)$$

We normalize  $\hat{f}(\xi, \eta)$ ,  $\hat{g}(\xi, \eta)$  by dividing by  $\sigma$  and let

$$\hat{f}^*(\xi, \eta) = \hat{f}(\xi, \eta)/\sigma, \quad \hat{g}^*(\xi, \eta) = \hat{g}(\xi, \eta)/\sigma, \quad (11)$$

denote these normalized quantities. The function  $|\hat{f}^*(\xi, \eta)|$  is highly oscillatory, and  $0 \leq |\hat{f}^*| \leq 1$ . The same is true for  $|\hat{g}^*(\xi, \eta)|$ . Since  $f(x, y)$ ,  $g(x, y)$  are real, their Fourier transforms are conjugate symmetric. In particular, the functions  $|\hat{f}^*(\xi, 0)|$  and  $|\hat{g}^*(\xi, 0)|$  are symmetric about the origin on the line  $\eta = 0$  in the Fourier transform plane.

For digitized  $512 \times 512$  images  $f(x, y)$ , the above smoothing is accomplished using discrete Fourier transforms via FFT algorithms, [2]. The discrete Fourier transform of  $f(x, y)$  is a  $512 \times 512$  array of complex numbers, which we again denote by  $\hat{f}(\xi, \eta)$  for simplicity. Here, the ‘frequencies’  $\xi, \eta$  are integers, which we assume reordered so as to lie between  $-256$  and  $256$ , with the zero frequency at the center of the transform array. This ordering is assumed



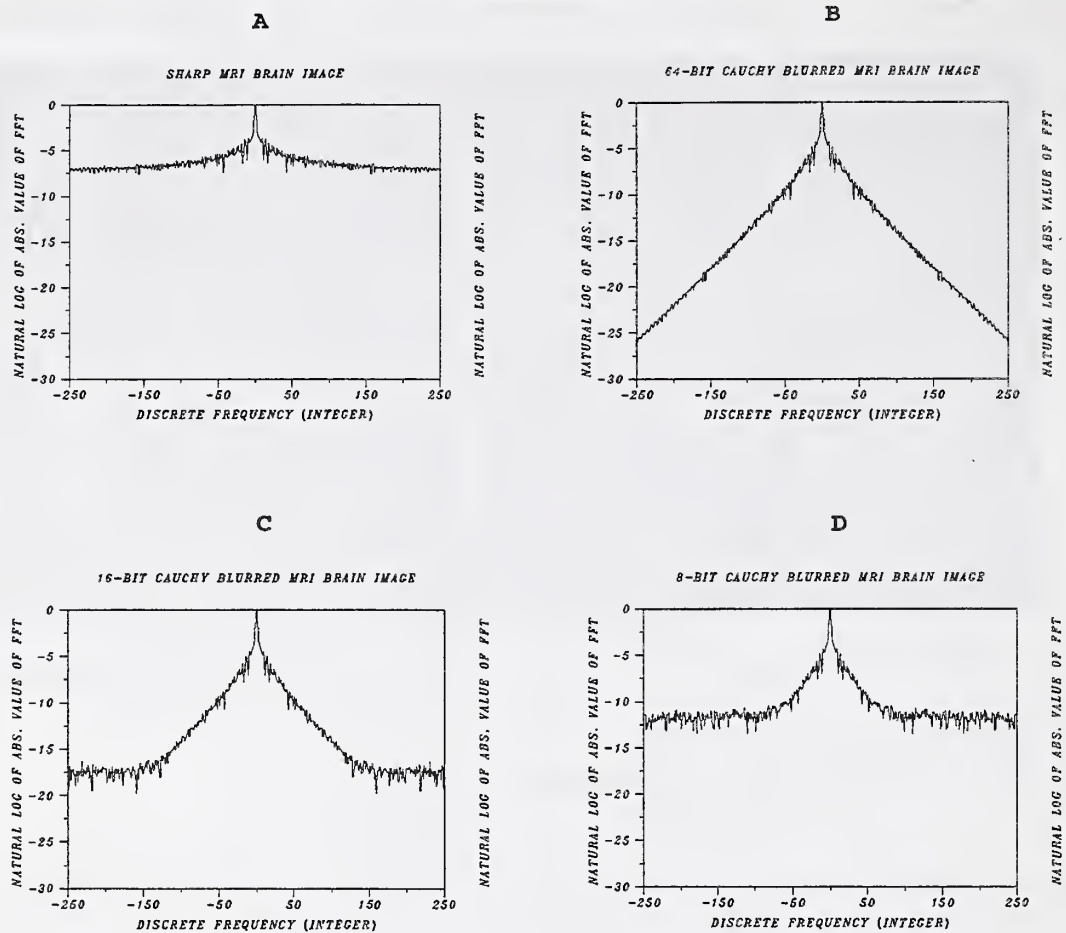


Figure 7: Blurring and truncation process in Fourier domain.

- A.  $\log |\hat{f}^*(\xi, 0)|$  in sharp MRI brain image.
- B.  $\log |\hat{g}_{64}^*(\xi, 0)|$  in Cauchy blurred MRI image stored in 64-bit precision.
- C.  $\log |\hat{g}_{16}^*(\xi, 0)|$  in Cauchy blurred MRI image stored in 16-bit precision.
- D.  $\log |\hat{g}_8^*(\xi, 0)|$  in Cauchy blurred MRI image stored in 8-bit precision.

Rounding noise in C and D masks important high frequency information. This explains why blurring followed by truncation is not fully reversible.

throughout this paper. With the same interpretation for  $\xi, \eta$  in (8), we multiply  $\hat{f}$  by  $\hat{h}$ , and inverse FFT the result to obtain the digitized smooth version  $g(x, y)$ . The normalized discrete transforms  $|\hat{f}^*(\xi, 0)|$  and  $|\hat{g}^*(\xi, 0)|$  have the same properties as their continuous counterparts. These 1-D traces of the 2-D FFT are easily visualized, and are helpful in understanding the blurring process.

## 4 Blurring and truncation in the Fourier domain

We now illustrate the foregoing by applying it to a digitized sharp image. The image  $f(x, y)$  in Fig. [6A] is a  $512 \times 512$  MRI sagittal brain image quantized at 8-bits per pixel. Thus, each pixel value is an integer lying between 0 and 255. The sharp image  $f(x, y)$  was blurred computationally by convolution with a Cauchy density. A double precision calculation was used to form

$$\hat{g}(\xi, \eta) = \hat{f}(\xi, \eta)e^{-0.075(\xi^2 + \eta^2)^{1/2}}, \quad (12)$$

where  $\xi, \eta$  are integer frequencies and  $-256 \leq \xi, \eta \leq 256$ . The inverse transform of  $\hat{g}(\xi, \eta)$  when calculated and stored in 64-bit precision will be denoted by  $g_{64}(x, y)$ . We may also form  $g_{16}(x, y)$  by truncating and storing in 16-bit precision, and  $g_8(x, y)$ , corresponding to truncating to 8-bit precision. These three blurred images are distinct but visually indistinguishable. The blurred image  $g_8(x, y)$  is shown in Fig. [6B]. Clearly, all the sharp features in the original image have been attenuated. Evidently, compression of image [6B] should be easier than compression of image [6A].

It is instructive to study these four images in the Fourier domain by displaying 1-D traces. First, consider the original sharp image  $f(x, y)$ . In Fig. [7A], we display  $\log |\hat{f}^*(\xi, 0)|$  on the interval  $-250 \leq \xi \leq 250$ . As expected, the trace is oscillatory and symmetric about the origin. Next, we examine the high precision blurred image  $g_{64}(x, y)$ . In Fig. [7B],  $\log |\hat{g}_{64}^*(\xi, 0)|$  is plotted on  $-250 \leq \xi \leq 250$ . Due to the negative exponential in (12), the trace in Fig. [7B] decays rapidly as  $|\xi|$  increases. This is again as expected. An important limitation of blurring begins to emerge when we consider the 16-bit image  $g_{16}(x, y)$ . In Fig. [7C],  $\log |\hat{g}_{16}^*(\xi, 0)|$  is plotted on  $-250 \leq \xi \leq 250$ . Comparison with Fig. [7B] reveals the effects of truncating to 16-bits prior to storing the blurred image. In Fig. [7C], the signal amplitude at high frequencies falls below 16-bit rounding noise. Therefore, in reconstructing  $f(x, y)$  from  $g_{16}(x, y)$ , we do not have available *all* of the necessary information. For this reason, blurring followed by truncation is *never* fully reversible. Truncation to 8-bits destroys considerably more information. In Fig. [7D],  $\log |\hat{g}_8^*(\xi, 0)|$  is plotted on  $-250 \leq \xi \leq 250$ . Comparison of Figs. [7B] and [7D] shows that signal amplitude at frequencies beyond  $|\xi| = 50$  falls below 8-bit rounding noise. Thus, any reconstruction of  $f(x, y)$  from  $g_8(x, y)$  must be based on the partial information contained in the narrow frequency range  $0 \leq |\xi| \leq 50$ . Of necessity, any such reconstruction cannot be expected to match the resolution and visual quality in the original  $f(x, y)$ .

## 5 Image deblurring

Image deblurring is a prime example of a numerically unstable problem. Small errors or noise in the blurred image data can become amplified if the deblurring procedure is not carefully designed. In particular, for the class of blurs described in (8), there is the potential for *explosive* error amplification. Consequently, the presence of 8-bit rounding noise in the blurred image, even though it is invisible, is a matter of concern.

There is a large literature on image deblurring methods. The books [1] and [9] are classics in the field. Many deblurring methods are based on probabilistic considerations and are iterative in nature. An instructive survey of such methods is given in [14]. Unfortunately, for

large size images, thousands of iterations are usually necessary to resolve fine detail, [5]. Such iterative methods are therefore not appropriate in the present context of image compression. The method used in this paper is the recently developed fast SECB method [3], [4]. This is a direct non-iterative procedure, based on FFT algorithms. The method is designed for images with edges. Noise amplification is suppressed by proper *regularization*. This requires user selection of appropriate values for the regularization parameters, based on *a-priori* knowledge of what the true image should look like. As noted in [5], such prior knowledge is a fundamental requirement in all reconstruction methods.

Given the blurred image  $g(x, y)$  and the blurring kernel  $h(x, y)$ , the SECB method obtains the deblurred image  $f^\dagger(x, y)$  in the Fourier transform domain, through the following formula. With  $\bar{z}$  denoting the complex conjugate of  $z$ ,

$$\hat{f}^\dagger(\xi, \eta) = \frac{\bar{\hat{h}}(\xi, \eta) \hat{g}(\xi, \eta)}{|\hat{h}(\xi, \eta)|^2 + K^{-2} |1 - \hat{h}^s(\xi, \eta)|^2}, \quad (13)$$

leading to  $f^\dagger(x, y)$  upon inverse transforming. In (13),  $s$  and  $K$  are positive constants that serve as regularization parameters. The value of  $s$  should be fixed in the range  $0.001 \leq s \leq 0.01$ . The optimal value for the parameter  $K$  is a function of the noise content in the blurred image. In the case of the 64-bit blurred image  $g_{64}(x, y)$  discussed in Fig. [7B], a value of  $K = 100$  would be appropriate, while a value of  $K \approx 1$  is necessary to prevent noise amplification in the 8-bit blurred image  $g_8(x, y)$  in Fig. [7D]. With  $s$  fixed, the optimal value of  $K$  is best found interactively. Small values of  $K$  produce reconstructions with low resolution. Increasing  $K$  increases resolution, until a threshold value of  $K$  is reached. Further increases produce noisy reconstructions. The optimal  $K$  is the largest value that can be used before noise artifacts become unacceptable. For each choice of  $K$ , SECB deblurring of a  $512 \times 512$  image can be accomplished in under 5 seconds of CPU time on current desktop workstations.

## 6 Compression experiments

In the experiments discussed below, eight sample  $512 \times 512$  images were blurred with a Lévy density (8), with  $\alpha = 0.05$  and  $\beta = 0.4$ . The blurred images  $g(x, y)$  were truncated and stored as 8-bit images. These 8-bit images were deblurred with the SECB method (13), with  $s = 0.001$  and  $K = 0.833$ . This value of  $K$  produced small artifacts in some of the deblurred images. A lower value of  $K$  can be found that produces less visible artifacts, along with less resolution.

## 7 The results of the experiment

To compare the relative performance of the losslessly blurred technique versus jpeg we considered a set of relatively well known pictures and we proceeded as follows:

1. The original 512x512 grey scale image was blurred and truncated to eight bits.
2. The resultant image would be compressed near losslessly.



3. The original image would also be compressed, lossily, using jpeg at a quality setting that would match as closely as possible the compression ratio in step 2 above.
4. The original image, the blurred-truncated-deblurred version, and the compressed version would be constructed for subjective evaluation of the blur-truncate-deblur technique.

Typically, the blurred image would compress at 4:1 while the original would compress, near losslessly, at 2:1. But jpeg at about  $Q=90$  matches the compression ratio obtained by the lossless compression of the blurred-truncated version while maintaining superior quality ( $Q$  is a jpeg quality knob that varies between 0 and 100; its default value is 75 and values above 95 are NOT recommended for normal use because they severely degrade performance for marginal, if any, quality gains). Indeed, the blurred-truncated-deblurred images display high frequency systematic components which are noticeable in the light-colored and edge free parts of the image (e.g., the sky portion GoldHill). Therefore, it is obvious that the scheme we investigated does not compete with straightforward jpeg compression.

The sample images, shown below, are:

- Barbara: The blurred image compresses losslessly to 77K comparable to 76K of the jpeg version at  $Q=92$ .
- Boat: The blurred image compresses losslessly to 70K comparable to 68K of the jpeg version at  $Q=92$ .
- GoldHill: The blurred image compresses losslessly to 75K comparable to 72K of the jpeg version at  $Q=90$ .
- Lena: The blurred image compresses losslessly to 69K comparable to 65K of the jpeg version at  $Q=92$ .
- Mandrill: The blurred image compresses losslessly to 95K comparable to 93K of the jpeg version at  $Q=86$ .
- Peppers: The blurred image compresses losslessly to 71K comparable to 67K of the jpeg version at  $Q=91$ .
- WashSat: The blurred image compresses losslessly to 62K comparable to 59K of the jpeg version at  $Q=90$ .
- Zelda: The blurred image compresses losslessly to 65K comparable to the 62K of the jpeg version at  $Q=93$ .

The images that follow are printed three at a time with the top image being the original, the bottom left the blurred-truncated-deblurred version and the bottom right the j-peg version, after reconstruction, whose compression ratio matches the compression ratio of the blurred image when compressed losslessly. We also include as separate full images the three versions of GoldHill because the image contains several flat portions over which one can clearly identify the systematic distortion introduced through truncation. Although not immediately visible in the other images, the same type of image distortion is detectable when the images are magnified.



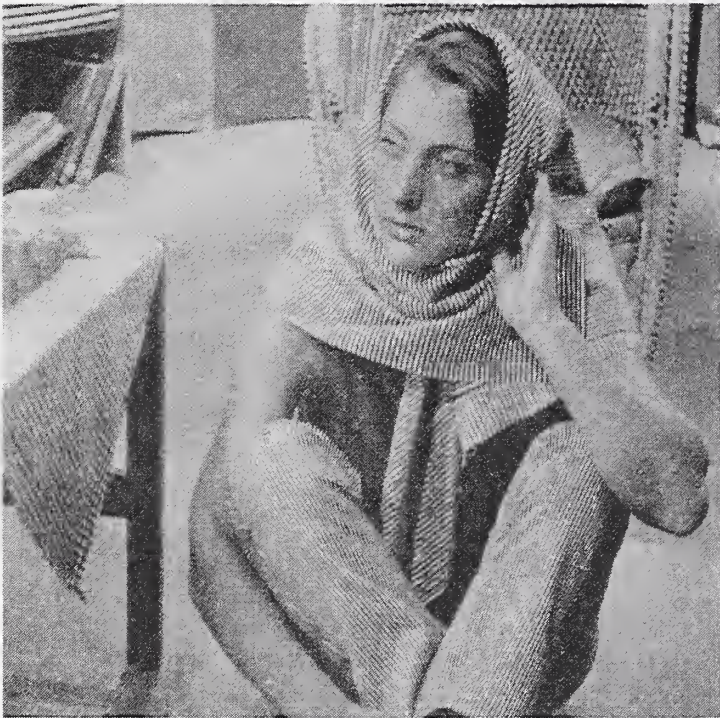
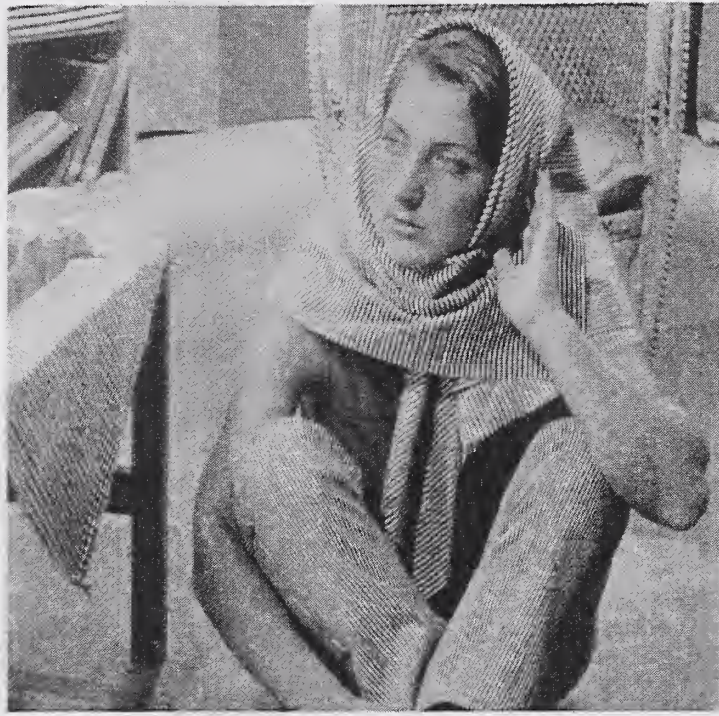


Figure 8: Barbara





Figure 9: Boat





Figure 10: GoldHill





Figure 11: GoldHill, Original





Figure 12: GoldHill, blurred-truncated-deblurred





Figure 13: GoldHill, jpegged, Q=90





Figure 14: Lena





Figure 15: Mandrill



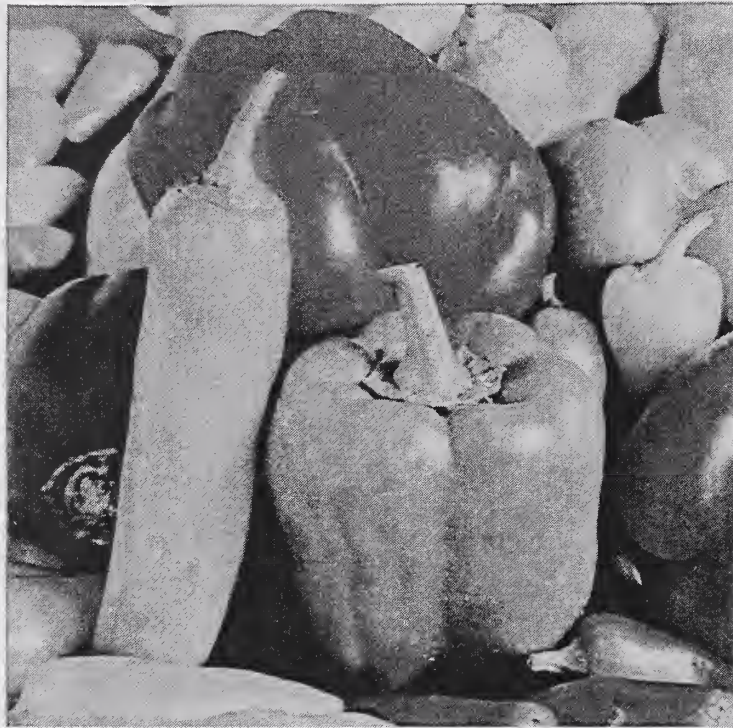


Figure 16: Peppers



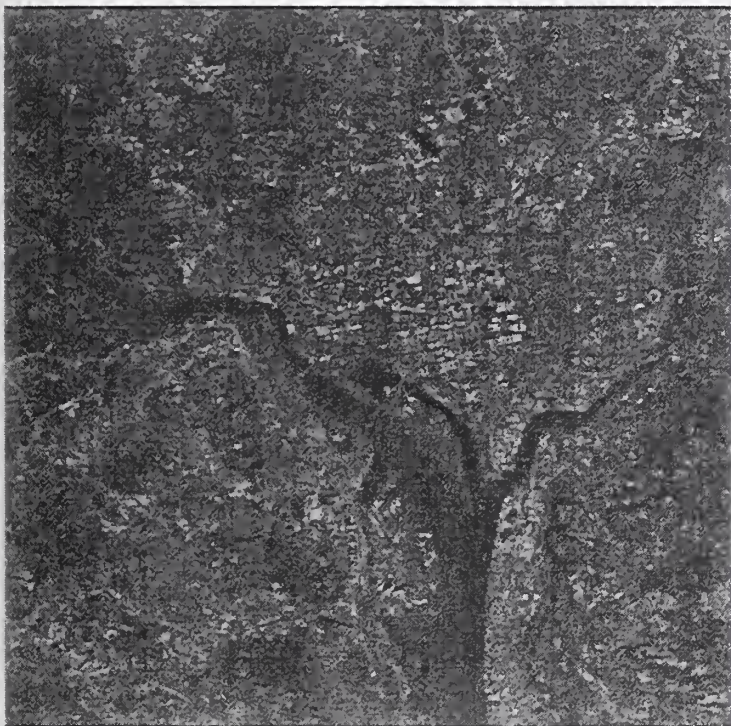


Figure 17: Washington by satellite





Figure 18: Zelda



# References

- [1] H. C. ANDREWS AND B. R. HUNT, *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [2] E. O. BRIGHAM, *The Fast Fourier Transform*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [3] A. S. CARASSO, *Image restoration and diffusion processes*, SPIE Proceedings, 2035 (1993), pp. 255–266.
- [4] A. S. CARASSO, *Overcoming Hölder continuity in ill-posed continuation problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1535–1557
- [5] A. S. CARASSO, *Linear and nonlinear image deblurring: a documented study*, SIAM J. Numer. Anal., 36 (1999), pp. 1659–1689.
- [6] H. S. CARSLAW, *Introduction to the Theory of Fourier Series and Integrals*, Dover, 1930.
- [7] W. FELLER, *An Introduction to Probability Theory and its Applications*, Vol. 2, Second ed., Wiley, New York, 1971.
- [8] PHILLIP FRANKLIN, *An Introduction to Fourier Methods and the Laplace Transformation*, Dover 1958 (1949).
- [9] R. C. GONZALEZ AND P. WINTZ, *Digital Image Processing*, Second ed., Addison-Wesley, Reading, MA, 1987.
- [10] BRAD JOHANSON, *Optimizing Perceptual Quality in JPEG Coded Images*, <http://www-ise.Stanford.EDU/class/psych221/98/jpeg/brad/> (1998)
- [11] JONATHAN KNUDSEN, *JAVA 2D Graphics*, O'Reilly, 1999.
- [12] CORNELIUS LANZOS, *Discourse on Fourier Series*, Oliver and Boyd, 1966.
- [13] DOUGLAS A. LYON, *Image Processing in Java*, Prentice Hall, 1999.
- [14] E. S. MEINEL, *Origins of linear and nonlinear recursive restoration algorithms*, J. Opt. Soc. Am. A, 3 (1986), pp. 787–799.
- [15] TERRY M. PETERS AND JACKIE WILLIAMS, *The Fourier Transform in Biomedical Engineering*, Birkhuser, 1998.
- [16] GILBERT STRANG AND TRUONG NGUYEN, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.
- [17] I. TERNOVSKIY AND T. JANNSON, *Novel Video Processing Schemes Integrating Image Compression and Pattern Recognition*, #3164-20, SPIE 42nd meeting; San Diego, 1997.
- [18] JAMES S. WALKER, *Fourier Analysis*, Oxford University Press, 1988.







